# Utilizing Dynamic Programming to Aid in the Hybrid Electric Vehicle (HEV) Component Selection Process to Minimize the Vehicle's Fuel Consumption

Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Bachelor of Science in the College of Engineering at The Ohio State University

By

Kristina Kuwabara

The Ohio State University

2019

Thesis Committee

Dr. Shawn Midlam-Mohler, Advisor

Dr. Giorgio Rizzoni, Advisor

Abstract

The EcoCAR Mobility Challenge is a four-year competition sponsored by General Motors, Department of Energy, and MathWorks that challenges 12 universities to transform a conventional 2019 Blazer into a semi-autonomous connected and hybrid vehicle. During the first year of the competition, the team determines the vehicle architecture that meets three engineering goals: reducing fuel consumption, maintaining stock vehicle drive quality, and ensuring a minimum acceleration performance.

To determine the appropriate vehicle architecture, a design search was performed that utilized various simulations platforms to narrow the design space to one solution. This research focuses on the usage of dynamic programming as a tool to properly size components with regards to increasing the vehicle's fuel economy. For a multiple-stage decision making process, dynamic programming (DP) minimizes a cost function through backward calculation over a sequence of decisions. For this applied research, DP computes the optimal control variables associated with the hybrid torque split and transmission gear state at each specific time step of a drive cycle. Also, DP eliminates control variable combinations that cause components to operate in an infeasible way. The advantage of DP is the determination of the global minimum fuel consumption, which guarantees the component configuration are fairly evaluated against one another. This research goes over the process and results of sizing a

hybrid’s energy storage system, front powertrain system, electric motor, and rear final drive ratios.

This document is dedicated to my parents that have always supported me and given me the belief that anything is achievable with hard work and dedication.

# Acknowledgment

I would like to thank my advisor, Dr. Shawn Midlam-Mohler for his guidance throughout EcoCAR and my undergraduate career. His technical expertise and mentorship have aided in this research as well as showing me a career path that I'm excited to pursue. I'd also like to thank Dr. Giorgio Rizzoni for always being willing to provide support and insight throughout this project. Next, I want to thank all the past EcoCAR team leads that have prepared me to be successful in this competition. Lastly, I want to thank all the current EcoCAR team members for always being fun and supportive; they have made working on this project a life changing experience.

Table of Content

List of Tables

List of Figures

# Chapter 1. Introduction

Throughout the world, the automotive industry is quickly advancing with recent focus in creating a more sustainable and safer transportation solution by using electrification, connected, and automated technology. For the past 28, the US Department of Energy has partnered with the North America Auto-Industry to oversee the advanced vehicle technology competitions (AVTCs). The current integration of the AVTC program is the EcoCAR Mobility Challenge, which challenges 12 different universities to reengineer a 2019 Blazer to have a high fuel economy, implementation of connected and automated technology, and target the mobility-as-a-service market. Students will emulate an industry-like vehicle design process over a four-year period outlined in Figure 1.

The program is currently in its design year with a primary focus of choosing components that will meet the high-level engineering goals of increasing the fuel economy, maintaining conventional vehicle drive quality, and keeping the expected acceleration performance.

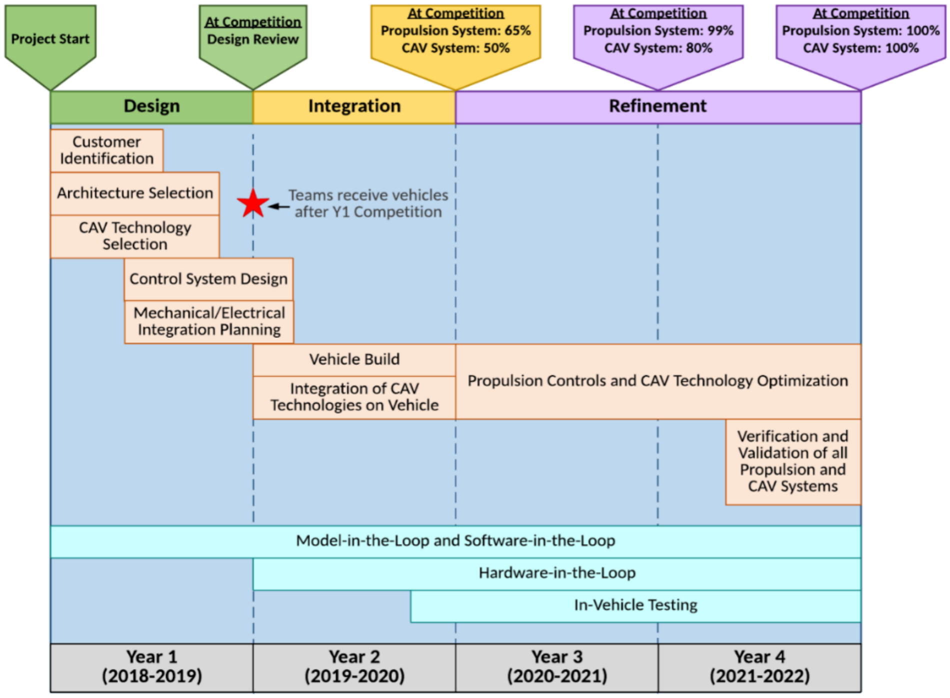


Figure 1: Vehicle Design Process

## 1.1. Vehicle Architecture Selection Process Overview

To determine the vehicle architecture selection, a 4-stage process was followed to ensure that the entire design space was surveyed. This process was summarized in Figure 2.

**Stage 1: Limiting the Design Space**

**Stage 2: Determining the Vehicle Configuration**

**Stage 3: Elimination of Components that Fail to Meet Team's Acceleration Requirement**

**Stage 4: Determining the Component Sizes**

| Stage | Vehicle Model/ Environment | Purpose |
|---|---|---|
| 1 | GREET | • Determine Fuel Type<br>• Determine Hybrid Type |
| 2 | Autonomie | • Fuel Economy Simulation |
| 3 | Autonomie | • Acceleration Performance Simulation |
| 4 | Vehicle Kinematic Model for Dynamic Programming | • Fuel Economy Simulations |
| | Autonomie | • Validation of Dynamic Programming Results |

Figure 2: OSU Vehicle Architecture Selection Process

Throughout the vehicle architecture selection process, multiple vehicle models or simulation environments were used to eliminate options that would not allow the team to optimize their engineering goals. The first stage utilized the Argonne National Lab software GREET to help make the decision of the fuel type and hybrid type. Next, the Argonne National Lab software Autonomie was used to perform initial fuel economy simulations to determine if there was an advantage for a specific vehicle configuration, specifically with respect to the motor placement. From this step, the team decided to go with a P0-P4 Hybrid Electric Vehicle (HEV), since other motor placements did not have a large enough advantage to outweigh the high integration risk of a P1, P2, or P3 motor. The next stage used Autonomie to eliminate configurations of different components that would not meet the team's minimum acceleration requirement. Then, the last step was to determine the component sizes that would maximize fuel economy. Within this stage, the team used dynamic programming, which is a multi-step decision making process to minimize a cost function. For this application, dynamic programming minimized the fuel consumption and ensured optimal hybrid controls to give confidence that all component combinations were fairly evaluated. Furthermore, as this competition focuses on integration of connected and automated vehicles, these fuel economy results represent the optimization capabilities with look-ahead information, such as vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication.

## 1.2. Research Motivation

The usage of dynamic programming aided in the component selection process. During the component selection process, it was vital to ensure each vehicle architecture has optimal controls to ensure that the maximum ability of the vehicle's architecture was evaluated.

During the last iteration of the AVTC competition, EcoCAR 3, the team solely used Autonomie for rapid modeling of vehicles architecture. The team calibrated the software's hybrid supervisory controller with a general algorithm that optimized thirteen control parameters to minimize fuel consumption, while ensuring drive trace error and State of Charge (SOC) deviations were within EPA standards.
However, the short coming of this generic algorithm was that the optimal controls were not guaranteed. Therefore, for the EcoCAR Mobility Challenge vehicle architecture evaluation process, dynamic programming would eliminate the calibration of component configuration specific controllers and allow for a guaranteed optimal control strategy that would minimize the fuel consumption.

## 1.3. Research Objectives

There were three major objectives of this research:

- To ensure that the model accurately represented the vehicle behavior with proper implementation of component data.
- To ensure that control variables acted in a feasible manner, and met the competition specified drive cycles.
- To determine the optimal size of a rear electric motor (REM), REM gear ratio, engine, transmission, and energy storage system (ESS) to minimize fuel consumption.

These objectives were completed through the creation of a MATLAB script-based look-up table model with the component data interchangeable between component configurations. The controls algorithm was ensured to be feasible based on the model set up to check for multiple infeasibilities during each evaluated step of the backward and forward calculations. If an infeasibility was found during the backward calculation, the control option was eliminated by assigning it an extremely high cost value, while if the controls acted in an infeasible manner during the forward calculation the dynamic programming problem set up would be changed. The optimal size of multiple components was determined from a design of experiment. Then, the results were analyzed to extract trends and create direct fuel economy component comparisons.

## 1.4. Chapters Overview

The rest of the thesis is outlined below that will complete the objectives of this research:

Chapter 2 discusses prior design space explorations for a hybrid electric vehicle. This chapter also discusses the concept of dynamic programming and prior examples of the usage of dynamic programming for vehicle architecture selection.

Chapter 3 goes over the setup of the dynamic programming problem, this will include the design of the P4 vehicle look-up table model, the chosen control variables, and the implementation of controller infeasibilities and drive quality penalties.

Chapter 4 summarizes the simulation and results that determined the optimal size of a Rear Electric Motor (REM), rear final drive ratio, front powertrain system, and energy storage system (ESS) to minimize fuel consumption.

Chapter 5 concludes the thesis by explaining key takeaways from this project and discusses the future research that can be built off this work.

# Chapter 2. Literature Review

When designing a vehicle, its vital to consider all available component and vehicle architectures. For the past 20 years, design space explorations have been used to determine the ideal design for a hybrid electric vehicle. Compared to conventional vehicles, hybrids have more degrees of freedom with the addition of an electrified powertrain.

This chapter will discuss previous design space explorations for hybrid vehicles, as well as introduces the novel approach of dynamic programming and its ability to aid in the vehicle architecture selection process. Then, this chapter will conclude with analyzing previously performed case studies that used dynamic programming for vehicle architecture selection.

## 2.1. Design Space Exploration

In 1998, a design space exploration evaluated over 2.15 million hybrid architectures [1] that utilized a multi stage approach. This design space exploration eliminated vehicles that did not meet a specified value within the following areas: maximum acceleration, top speed, city-driving efficiency, and highway-driving efficiency. This approach captured the importance of meeting the general market trends. Then it utilized a dominance filter comparing hybrid designs against each other in the four-criteria area specified above. The criteria got stricter and harder to meet that eventually eliminated the choice down to 173 designs. Hybrid electric vehicles often want conflicting performance criteria such as high acceleration capability, low fuel consumption, and low vehicle cost. Throughout this process, there were tradeoffs made to ensure that each of the criteria were adequately met, which was defining the cost function. The final step of this design space exploration utilized humans to filter through the remaining design. Humans determined the final tradeoff by visually seeing how each design fit into various criteria. This design space exploration shows the importance of understanding the tradeoffs and a defined cost function.

Next in 2008, another design search was conducted that evaluated the preliminary design of hybrid vehicles with a two-step optimization method [2]. This first step of the optimization method was a multi-objective optimization. This process was narrowing down the designs to meet specific requirements. The second step was multi-criteria decision-making approach, which once again evaluated different criteria to ensure a tradeoff. For this analysis, more complex filtering techniques were utilized such as Hurwicz Algorithm that helped compromise between criteria. This analysis had nine different criteria evaluated, which was more advance than the 1998 design space exploration. In conclusion, the fundamentals of these design space exploration were similar. Designs were eliminated that did not meet overarching requirements than a tradeoff was made between different criteria. However, the more present-day study had a more sophisticated approach in analyzing the design capabilities and filtering the designs.

OSU EcoCAR architecture selection process has followed many other design space explorations by defining the basic requirements that must be met and eliminating architectures that do not meet these requirements. Criteria such as drive quality and performance were considered in early stages of the architecture selection process. To account for drive quality, components were eliminated that were not met to be used together. For example, the team eliminated a manual transmission to avoid drive quality concerns of creating a system that connected an engine and transmission that were not calibrated together. The initial stages also eliminated designs that would not meet the team's minimum acceleration and vehicle top speed requirement. Dynamic programming was utilized at the end of the design space exploration to compare the most important criteria of vehicle efficiency for the 80 final designs. At this stage, the cost function was based only on fuel consumption.

## 2.2. Dynamic Programming

To understand each architecture capability of minimizing fuel consumption, the optimization strategy known as dynamic programming was used. Dynamic programming is a numerical method for solving a multistage decision-making problem [3]. For this specific use case, the optimal controls determined each architectures' minimum fuel consumption for competition specified drive cycles to compare against each other. When utilizing DP, a major advantage is it does not require the formal calibration of a forward-looking controller; however, it is not real time implementable since it requires information for the entire optimization horizon. Since this design space exploration is done completely offline, dynamic programming is the appropriate tool to use.

DP is governed by the *Bellman's principle of optimiality* that states an optimal policy has the property that whatever the initial state and final decisions are, the remaining decision must constitute an optimal policy with regards to the state resulting from the first decision [4]. This can be summarized by stating that to get from an initial to final state there is only one set of decisions that provide the optimal results.

A major use case for dynamic programming is the application to the energy management problem for hybrid vehicles. Hybrid vehicles have multiple torque sources that must be operated in an efficient manner to increase the vehicle's efficiency. The decisions of the power split between torque producing components, the gear state of the transmission, and the powerflow of the vehicle was optimized at each time step for an optimal fuel economy. The setup of this dynamic programming problem is explained in  Chapter 3.

## 2.3. Simplified Dynamic Programming Example

A simplified problem was created to explain the concepts of dynamic programming. For this problem, a conventional vehicle with properties shown in Table 1 was required to meet a basic drive cycle found in Figure 3.

Table 1: Conventional Vehicle Properties for DP Example Problem

| **Vehicle Parameters** | **Parameter Value** |
|---|---|
| Transmission 1st Gear | 2.66 |
| Transmission 2st Gear | 1.78 |
| Transmission 3st Gear | 1.30 |
| Transmission 4th Gear | 1.00 |
| Transmission 5th Gear | 0.80 |
| Transmission 6th Gear | 0.64 |
| Engine Redline Speed | 6000 RPM |
| Final Drive Ratio | 3.75 |
| Wheel Radius | 12 inches |

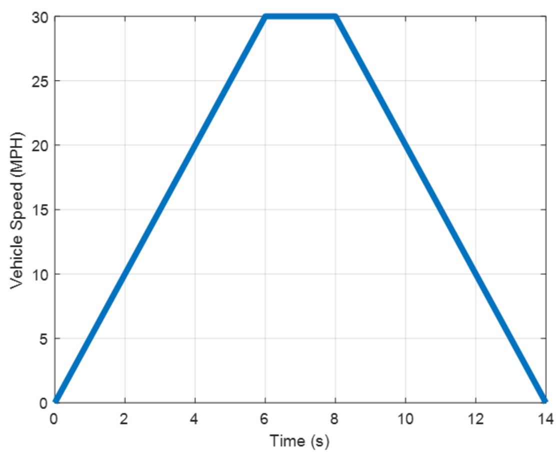


Figure 3: Drive Cycle for DP Example Problem

For a conventional vehicle to meet the drive trace, the engine must fulfill the driver torque request leaving the only unconstrained variable as the gear state of the transmission. A grid of the transmission gear options and the required speed at each time step can be found in Figure 4. The conventional vehicle has a fixed initial and final state highlighted with the blue box, since an operating engine at 0MPH must be in neutral and disconnected from the wheels.

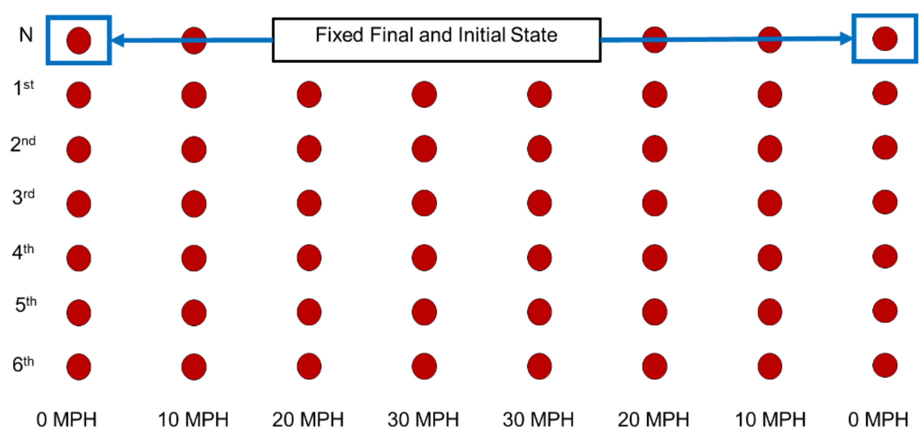


Figure 4: Grid Space Showing Transmission Gear vs Drive Trace

To determine the minimum fuel cost, the team backwards calculated the optimal gear trajectory for the simplified drive cycle shown in Figure 5. The cost to go is represented by a fuel cost or an infeasibility cost. An infeasible gear was assigned a high cost to that specific path to ensure the it was never chosen. When going from 0 MPH to 10 MPH, 2nd through 6th gear was infeasible due to drive quality concerns and engine stall limits.

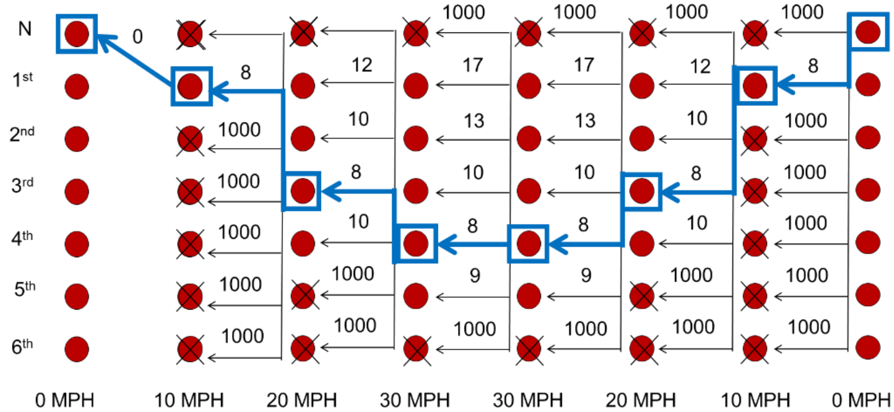


Figure 5: Optimal Gear Trajectory for DP Example Problem

This backward calculation approach was utilized to determine the optimal controls for each hybrid vehicle design to evaluate their minimum fuel consumption. For a hybrid vehicle, there was additional degrees of freedom, which included the torque split between propulsion components, gear state, and specified power flow.

## 2.4. Previous Case Study with Dynamic Programming

Dynamic programming has been applied to previous design space exploration for an electric deliver truck [5]. For this design search, series plugin hybrid electric delivery trucks were evaluated that had various power flows. All powertrains were based on the same model structure changing the way powertrain components were connected. For this case study, each component was modeled with simple governing equations or look up tables. A similar approach for the creation the vehicle model was used for this research. The electric deliver truck was analyzed for the NREL PG&E Utility Truck drive cycle, NREL Baltimore Parcel Delivery drive cycle, and Manhattan drive cycle. One of the major conclusions of this cases study was the characteristics of the driving mission had a significant effect on how the powertrain was sized, and to have a more robust analysis a multitude of driving cycles should be considered [5]. The final differential ratio was one property that was highly dependent on the specific drive cycle.  For the proposed research, two competition provided drive cycles that emulated city and highway driving were used to size component. This ensures the system was designed for a variety of use cases.

## 2.5. Chapter Summary

This chapter explained the concept of dynamic programming and evaluated various design space exploration for hybrid vehicle. The research will build off of past knowledge to determine the optimal hybrid architecture for a new set of requirements based on the EcoCAR Mobility Challenge.

To run this optimization problem, there were three algorithms required: a simulation initialization file, the vehicle model, and a DP solver. The major functions of each of these algorithms are outlined in Figure 6.

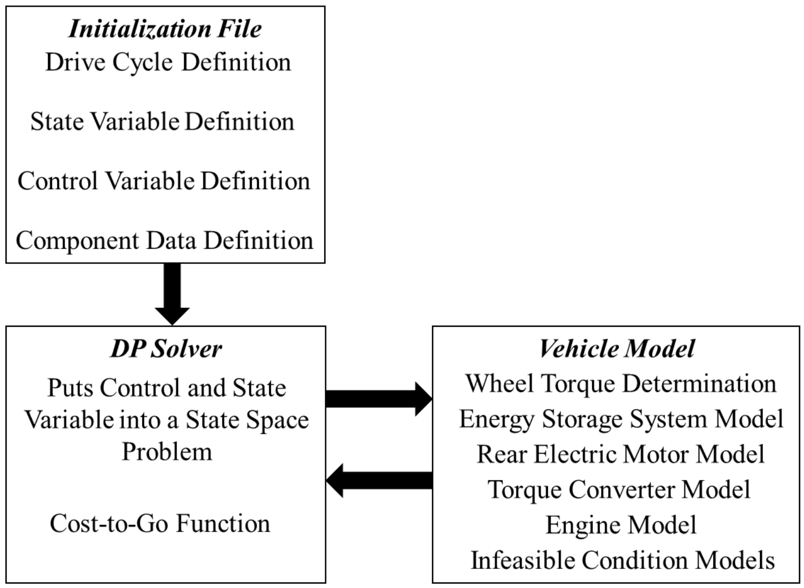


Figure 6: Major Algorithm Relationship for Dynamic Programming Problem

The code execution path begun with the initialization file that defined the simulation parameters, such as the component data, the control variables, and the state variables. Then the DP solver and vehicle model were called upon and worked in conjunction to backward calculate the optimal controls that would minimize the cost function. These algorithms were set up with the simulation purpose of evaluating different component combination for a P4 vehicle architecture shown in Figure 7.

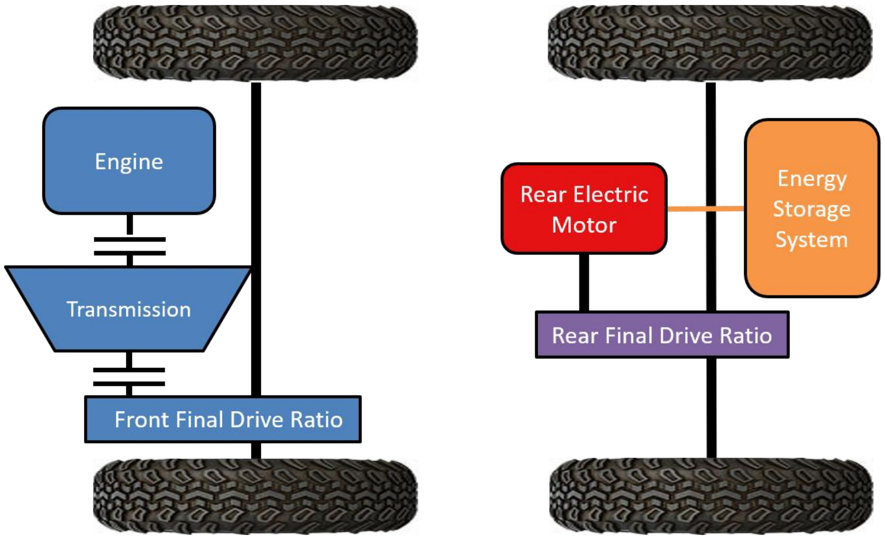


Figure 7: P4 Vehicle Architecture

### 3.1. Initialization File

The initialization file defined the problem's control variables, state variables, model data, and drive cycles.

#### 3.1.1. Control Variable Definition

At each instance of driving a hybrid vehicle, decisions must be made on how to operate the different components to achieve performance metrics. Within these simulations, the variables that were controlled and optimized were the electric motor torque, engine torque, gear state, and mode operation over a specified drive cycle. The weighting of electric motor torque and engine torque, known as the hybrid power split, continues to be a major controls challenge of hybrid vehicle development. Therefore, this problem determined the optimal power split to meet the drive cycle, while minimizing fuel consumption. When evaluating the gear state as a control variable, the engine has the ability to be at a specific efficiency point and have different wheel torque available, which adds a degree of freedom to the simulation.

There were two modes that the vehicle could operate in an electric-mode and a parallel mode shown in

Figure 8 and

Figure 9. The full electric-mode met the acceleration request with the motor or decelerated the vehicle through regenerative braking. There were two possible conditions for parallel mode: the engine and electric motor worked together to meet the drivers request or the engine and electric motor exceeded the drivers request and excess torque would charge up the battery due to coupling of the front and rear powertrain through the road.

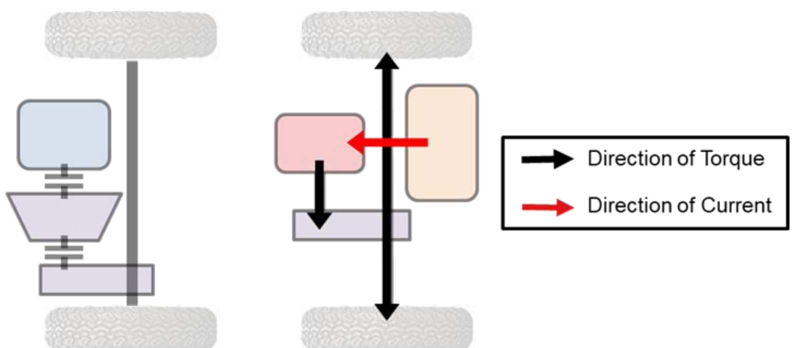


Figure 8: Full Electric Mode

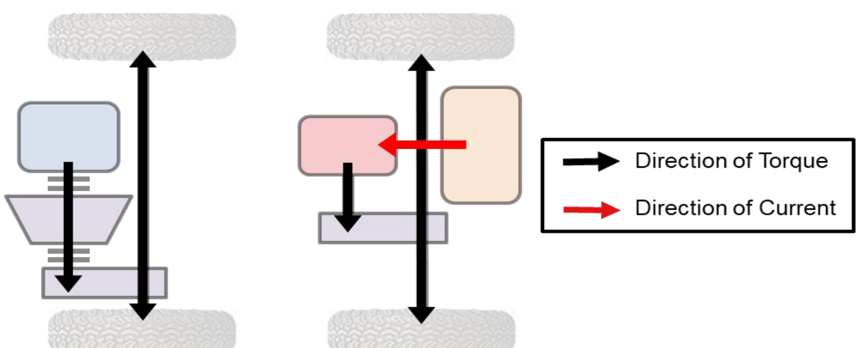


Figure 9: Parallel Mode

### 3.1.2. State Variable Definition

The state variables for this problem included the battery state of charge (SOC) and the mode of operation. The final and initial SOC have been constrained to ensure the vehicle operated in a charge sustaining state. The battery was limited to operate between 35% and 75%, since beyond this range the lithium-ion cell’s nominal voltage goes non-linear, leading to overvoltage or undervoltage of the system. The state of charge is a parameter manipulated by the charging or discharging of the battery, determined based on the control variable. Therefore, the optimal result will have a corresponding optimal SOC trajectory for a given drive cycle.

The mode of operation was included as a state variable to penalize engine restarts. During the backward calculation, the current time step and next time step modes were compared to penalize mode oscillations based on the energy required from a belted alternator starter (BAS) to restart the engine.

### 3.1.3. Discretization of Variables

The discretization of the continuous control and state variable affect the accuracy of the results. With finer control available of the engine and REM torque, the results converged to a minimum fuel consumption. Furthermore, when the SOC discretization was increased, it allowed for more coverage of the SOC trajectory and reduced the interpolation between steps. This will be further discussed in Section 4.1, which laid out the steps taken to ensure an appropriate discretization was used.

### 3.1.4. Model Data Implementation

The major goal of the design space exploration was evaluating a variety of motors, gear ratios, and engines. Each component’s data was set up in a standardized

MATLAB structure array. In the beginning of the initialization file, the various components under evaluation were selected, and simulations were ran that cycled through all possible component combinations.

3.1.5. Drive Cycle Implementation

Based on the EcoCAR Mobility Challenge fuel economy testing requirements, two custom drive cycles were evaluated that will be referred to as EMC City and EMC Highway, which can be found in Figure 10. The competition has a specific weighting of these drive cycles that will be used to determine an overall fuel economy number found in Equation 1.

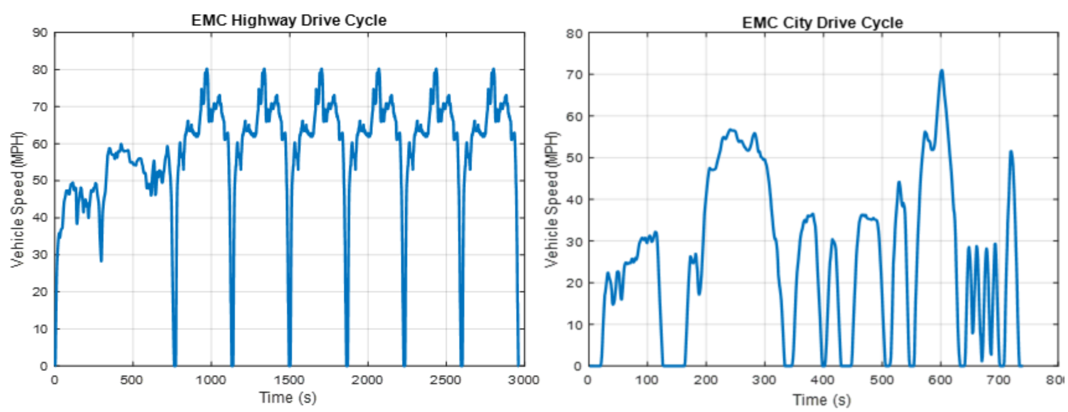


Figure 10: EMC City and EMC Highway Drive Cycles

$$FE_{EMC,\,Combined} = \frac{1}{\frac{0.55}{FE_{EMC,\,City}} + \frac{0.45}{FE_{EMC,Highway}}} \quad [1]$$

3.2. DPM Function

The DPM function was developed by ETH Zurich as a generic algorithm to be utilized to solve discrete-time optimal control problems [6]. The DPM function transformed the initialization file into a multi-dimension state space problem, that backward calculated the minimum cost, in this case fuel consumption, to meet the drive trace.

At each time step and SOC state, a minimum cost to go $(J_{k \to N})$ from the current time step (k) to the final time step (N) was determined based on Equation 2 that can be visualized in Figure 11. The vehicle model evaluated every control variable combination at each discrete SOC state to determine the minimum arc costs $(J_{k \to k+1})$. The total cost $(J_{k+1 \to N})$ represented the cost of going from the previous state to end of the drive cycle. This was often interpolated between two fixed states cost. This interpolation could cause a cumulative error if too low of SOC discretization was chosen. This overall minimum cost was calculated when the initial state was reached (k=1).

$$J_{k \to N} = J_{k \to k+1} + J_{k+1 \to N} \qquad [2]$$

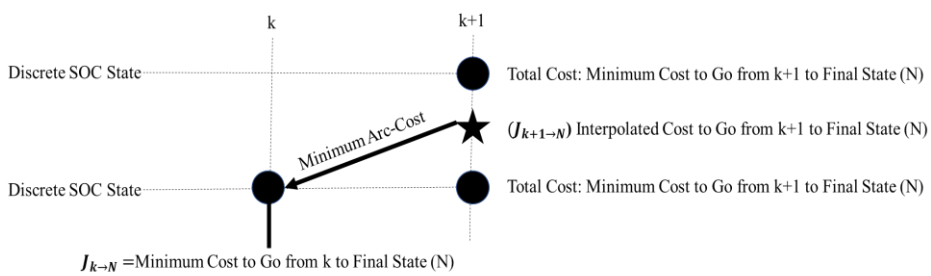


Figure 11: Visualization of Cost to Go Function at Each Time Step

Since every control variable combination was evaluated, each simulation checked that the control inputs allowed the components to act within their ability. The model checked for multiple infeasibilities explained in Section 3.3.7. The arc cost determined by Equation 3 was calculated based on fuel consumed C{X} and by assigning an infinite like cost penalty $Inf_{cost}$ that would eliminate any infeasibilities (I).

$$J_{k \to k+1} = C\{X\} + I \times Inf_{cost} \quad [3]$$

### 3.3. Vehicle Model

The vehicle model was a low fidelity steady state look up table-based model that was governed by vehicle kinematic equations. The vehicle model begun with the calculation of the wheel torque request, then had simplified model of the energy storage system, rear electric motor, torque converter, transmission, engine, and component infeasible. The vehicle model was in a MATLAB script form and was solved for a multidimensional matrix that considered all possible control and state variable combinations.

#### 3.3.1. Wheel Torque Request

The wheel torque request was based on Equation 4, which considers the simplified longitudinal vehicle dynamics. The wheel torque request was a combination of the aerodynamic drag, rolling resistance, and acceleration request of the drive cycle.

$$T_{wheel} = (F_{drag} + C_d v^2 + m_{vehicle} a_x) \times r_{wheel} \quad [4]$$

#### 3.3.2. Energy Storage System

Next, the energy storage system was modeled to determine the maximum power limits of the electric motor. This was an essential step when evaluating the lower power battery option that constrained the electrification of the vehicle.  The maximum current discharge and charge limits were determined by Equation 5-Equation 6, respectively. This considered the system's nominal voltage and battery power limits. The maximum current value remained constant for the entire simulation.

$$I_{max,charge} = \frac{P_{max,charge}}{V_{Nominal}} \qquad [5]$$

$$I_{max,discharge} = \frac{P_{max,\ discharge}}{V_{Nominal}} \qquad [6]$$

Next, the maximum discharge and charge power were determined within Equation 7-Equation 8 based on the maximum current limits, battery efficiency, cell configuration, open circuit cell voltage, and cell resistance, which varied with state of charge.

$$P_{battery,charge} = \frac{N_s}{N_p} R_{cell} I_{max,charge}{}^2 + N_s V_{oc,cell} I_{max,charge} \qquad [7]$$

$$P_{battery,discharge} = \frac{N_s}{N_p} R_{cell} I_{max,discharge}{}^2 + N_s V_{oc,cell} I_{max,discharge} \qquad [8]$$

The maximum allowable propulsion or regenerative power eventually constrained the electric motor torque, to constrain the EM torque control variable to only consider the options that passed the battery pack constraints.

At the end of the vehicle model, the energy storage system was remodeled to govern the behavior of the SOC evaluated at each control variable combination based on Equation 9.

$$SOC(k+1) = \begin{cases} -\frac{1}{n_{coul}} \frac{I(k)}{Q_{nom}} + SOC(k) & \text{if } I(k) > 0 \\ -n_{coul} \frac{I(k)}{Q_{nom}} + SOC(k) & \text{if I(k)} < 0 \end{cases} \qquad [9]$$

3.3.3. Rear Electric Motor

The vehicle model determined the rear electric motor speed based on the rear final drive ratio and the vehicle speed, shown in Equation 10.

$$w_{EM} = \frac{v}{r_{wheel}} \times r_{rear,final\ drive} \qquad [10]$$

Then, the maximum REM torque demand was calculated within Equation 11 based on the wheel torque demand, the rear differential efficiency, and the rear final drive ratio, which corresponded to the electric motor filling the entire wheel torque request.

$$T_{REM,Max,Dmd} = \begin{cases} \dfrac{T_{wheel}}{r_{rear,final\ drive} \times \eta_{diff}} & \text{if } \ T_{wheel} > 0 \\ \dfrac{T_{wheel}}{r_{rear,final\ drive}} \times \eta_{diff} & \text{if } \ T_{wheel} < 0 \end{cases} \qquad [11]$$

Furthermore, the REM minimum and maximum torque limits were determined from the battery limits and the REM torque limits at a specific speed. Next, the electric motor torque was determined based on the vehicle operating mode (parallel/series), the control variable, the maximum electric motor torque demand, and REM torque limits. The electric motor torque command is summarized in Table 2.

When evaluating the parallel mode with a propulsive wheel torque demand, the control variable (U {1}) varied the electric motor torque from the minimum to maximum torque limits by the preset discretization.  If the vehicle was decelerating, the motor captured the maximum allowed regenerative torque within the component limits. Then if the vehicle was in EV Mode, the electric motor torque equaled the maximum electric motor torque demand. If the electric motor torque did not fall within the motors capabilities it would be eliminated, when checked for a component infeasibility.

Table 2: Rear Electric Motor Torque Determinations Governing Equations

| Mode | Maximum REM Torque Demand | Electric Motor Torque Determination |
|---|---|---|
| Parallel | $T_{REM,Max,Dmd} > 0$ | $T_{em} = \left(U\{1\} \times T_{em,max}\right)$ |
| | $T_{REM,Max,Dmd} \leq 0$ | $T_{em} = \begin{cases} T_{EM,Max,Dmd} & \text{if } -T_{EM,max} > T_{EM,Max,Dmd} \\ -T_{EM,max} & \text{if } -T_{EM,max} \leq T_{EM,Max,Dmd} \end{cases}$ |
| EV Only | $T_{REM,Max,Dmd} > 0$ | $T_{em} = \left(T_{EM,Max,Dmd}\right)$ |
| | $T_{REM,Max,Dmd} \leq 0$ | $T_{em} = \begin{cases} T_{EM,Max,Dmd} & \text{if } -T_{EM,max} > T_{EM,Max,Dmd} \\ -T_{EM,max} & \text{if } -T_{EM,max} \leq T_{EM,Max,Dmd} \end{cases}$ |

Based on the electric motor torque, the motor's efficiency was interpolated, and the required electric motor power was calculated from Equation 12. Since, the EM was only high voltage load on the system, the EM power represented the total electrical draw from the battery pack.

$$P_{em} = \begin{cases} \dfrac{T_{em} w_{em}}{\eta_{em}} & \text{if } \; T_{wheel} > 0 \\ T_{em} w_{em} \eta_{em} & \text{if } \; T_{wheel} < 0 \end{cases} \qquad [12]$$

3.3.4. Transmission

Next, the front powertrain of the vehicle was modeled. The transmission output shaft speed was determined based on Equation 13 that accounted for the vehicle speed, front drive ratio, and current gear state. The transmission gear was a control variable, that forced the evaluation of each gear ratio to determine the optimal gear state for every timestep.

$$w_{trans\,shaft,out} = \frac{v}{r_{wheel}} \times r_{front,final\,drive} \times \mathrm{r_{trans,gear}} \qquad [13]$$

3.3.5. Torque Converter

Next, the torque converter was modeled to manipulate the engine speed and engine torque values based on the torque converter's impeller and pump behaviors. At very low speed, the relationship between the vehicle speed and engine speed included the torque converter speed ratio to prevent the engine from stalling. The torque requirement from the impeller side of the transmission was calculated based on meeting the wheel torque request unfilled by the REM.  Then, the torque converter's pump torque otherwise known as the engine torque, a control variable, checked various possible engine torques to optimize the torque converters performance and keep the engine operating point in an efficient region.

3.3.6. Engine

The engine torque request was a control variable; however, its function was to determine the best way to utilize the torque converter and fulfilled the remaining wheel torque request with the front powertrain. When the vehicle operated in EV mode, the engine torque request was zero. The engine speed was based on the torque converter properties but remained to be influences by the vehicle speed.

The cost function of this problem was based on the fuel consumption determined through an engine fuel map. The fuel map utilized a spline interpolation to evaluate the arc cost to go from state to state.

3.3.7. Infeasibilities

When determining the optimal control strategy for a vehicle, the components must operate within their limitations. The model checked for numerous infeasibilities, and if an infeasibility was detected that control variable combination was assigned an infinite-like cost in the DPM function, therefore eliminated from consideration. The model checked for infeasibilities associated with the battery pack, electric motor, torque converter, and engine.

The infeasibilities associated with the battery pack performed a check to ensure the battery was operating within its charge and discharge limits.

The rear electric motor infeasibilities that were modeled included over speeding the motor or producing torque beyond the motor's capabilities at a specific speed.

Next, the torque converter was checked to ensure that the ratio between the impeller and pump torque did not exceed the maximum allowed torque ratio.

Lastly, the engine infeasibilities were modeled, including the engine redlining, stalling, or commanding torque beyond the engine's capabilities for a specific speed.

Through checking for infeasibilities, some control variable combinations were eliminated, but it ensured components operated within their capabilities.

### 3.3.8. Penalties

When optimizing a vehicle for fuel economy, often the results have frequent mode switching, which would be a major drive quality concern. Therefore, a penalty was implemented to decrease the vehicle's start of charge every time the engine was restarted. This penalty was determined based on the electrical power requirements to start-up an engine with a high voltage belted alternator starter.

## 3.4. Chapter Summary

This chapter outlined the necessary algorithms to set up the dynamic programming problem. The team developed an initialization file and a vehicle model that would be used in conjunction with a DP solver created by ETK Zurich. The vehicle model was low fidelity based on supplier look up tables and simple vehicle kinematic equations. These algorithms allowed various hybrid designs to be evaluated.

# Chapter 4. Simulations and Results

Once the dynamic programming problem was setup, simulations were performed to narrow the 80 component configurations to a finalized component configuration. This section will go through the process of determining the proper discretization required to ensure convergence of the optimal control solution. Then, this chapter outlines the simulations required, and trends associated with the selection of each component are presented.

## 4.1. Discretization Analysis

An analysis was performed to determine the required discretization for the continuous control and state variables: the EM torque, the ICE torque, and the battery SOC. Finer discretization resulted in the evaluation of more combinations of state and control variables at each time step. For simplicity, the EM torque and ICE torque were discretized at the same value. For this analysis, the torque discretization was varied from 10-200, while the SOC discretization was varied from 10-80. The relationship between the fuel consumption, torque discretization, and SOC discretization can be found in Figure 12. Based on the analysis, torque discretization had a larger impact on the convergence of a minimum fuel consumption than SOC discretization. For the remainder of the simulations, the SOC discretization was kept at 40, and the torque discretization was kept at 50. These discretization values ensured that the fuel consumption change by less than a 0.5%.

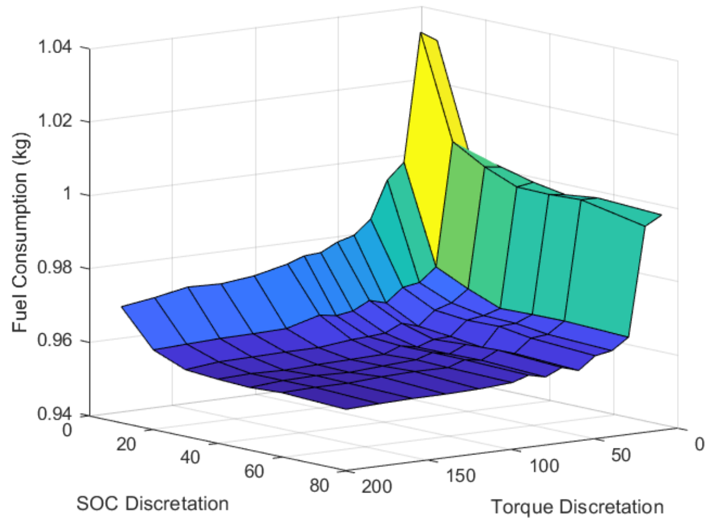


Figure 12: Torque and SOC Discretization vs Fuel Consumption

It was important to minimize the discretization of the control and state variable, while ensuring the solution converged. As the discretization was increased, the simulation time exponentially increased as seen in Figure 13.

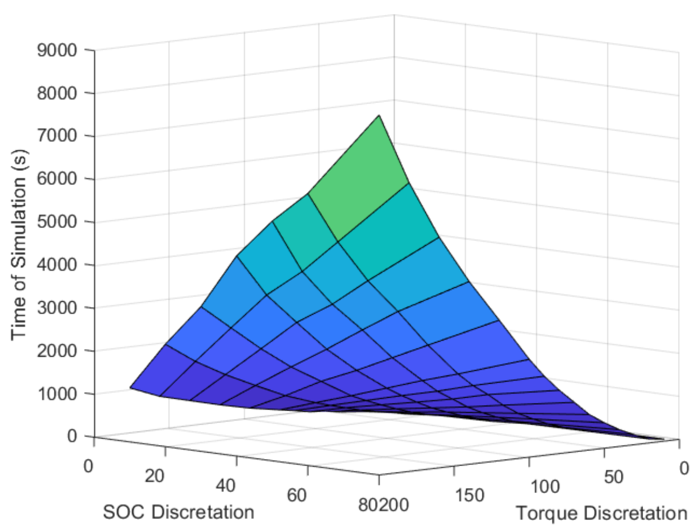


Figure 13: Torque and SOC Discretization vs. Time of Simulation

## 4.2. Required Simulation

There were a multitude of simulations runs to determine the best architecture to meet the competition requirements and achieve the highest fuel economy. The simulations evaluated 80 different configurations varying the components seen in Table 3 for the city and highway drive cycle. This process utilized the Ohio State Supercomputer [7] to supply the necessary computational power to run the 160 different simulations. This research utilized over 350+ hours on processors that were 48 or 28 cores.

Table 3: HEV Components Under Evaluation

| Components | Options |
|---|---|
| Energy Storage System (ESS) | 1.5 kW-hr ESS<br>3.5 kW-hr ESS |
| Front Powertrain System | GM LVG 1.5L Turbo with 3.49<br>GM LTG 2.0L Turbo with 2.89<br>GM LTG 2.0L Turbo with 3.17<br>GM LTG 2.0L Turbo with 3.8<br>GM LCV 2.5L NA with 3.47 |
| Rear Electric Motor (REM) | Parker Hannifin 90 kW REM<br>Parker Hannfin 112 kW REM |
| Rear Final Drive Ratio | 6.45<br>7.17<br>8.00<br>8.26 |

### 4.3. Results

The design space solved for the 80 different component configurations. The entire design space was evaluated to ensure that the chosen architecture was the most fuel efficient. Towever, this results section will go through the trends associated with the selection of each component.

#### 4.3.1. Energy Storage System

The energy storage system (ESS) had two options presented in Table 4.

Table 4: Energy Storage System Evaluated

| Component Properties | ESS Option 1 | ESS Option 2 |
|---|---|---|
| Cell Configuration | 1P80S | 8P96S |
| Peak Power Capabilities | 55 kW Discharge Peak Power 62 kW Charge Peak Power | 90 kW Discharge/ Charge Peak Power |
| Capacity | Total Energy: 1.5kW-hr Usable Energy:0.5 kW-hr | Total Energy: 3.5 kW-hr Usable Energy: 1.5 kW-hr |
| Nominal Voltage | 300 V | 348V |

The ESS was the component that showed the largest variation in fuel economy. The EMC city and EMC highway results can be found in Figure 14 and Figure 15, respectively. The percentages show the fuel economy improvement from ESS Option 1 to ESS Option 2. By displaying various front powertrain options, it demonstrates that the ESS trend affected the entire design space. For this simulation, the rear drive ratio and motor remained constant at 6.28 and 112-kW motor, respectively.

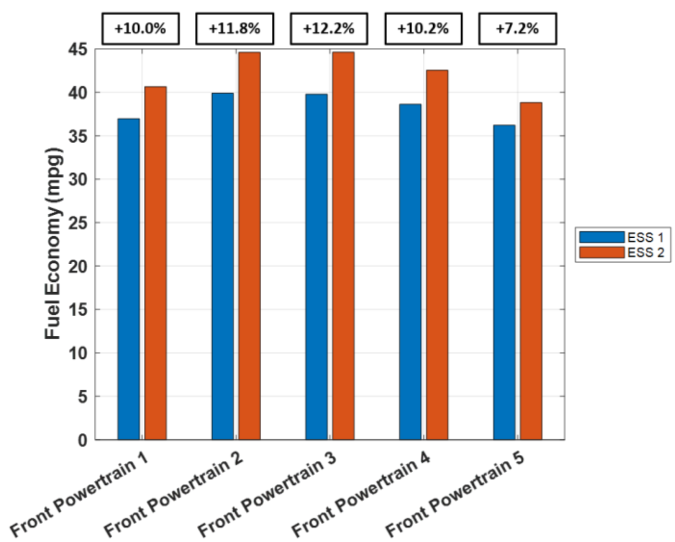


Figure 14: EMC City Fuel Economy Results for Varying ESS

For the EMC city drive cycle, fuel economy improvements between 7-12% were seen when choosing ESS Option 2 over ESS Option 1. Since, ESS Option 2 has a higher capacity and peak power, the hybrid vehicle had more occurrences where the engine was able to be turned off and be propelled with only electric propulsion components.

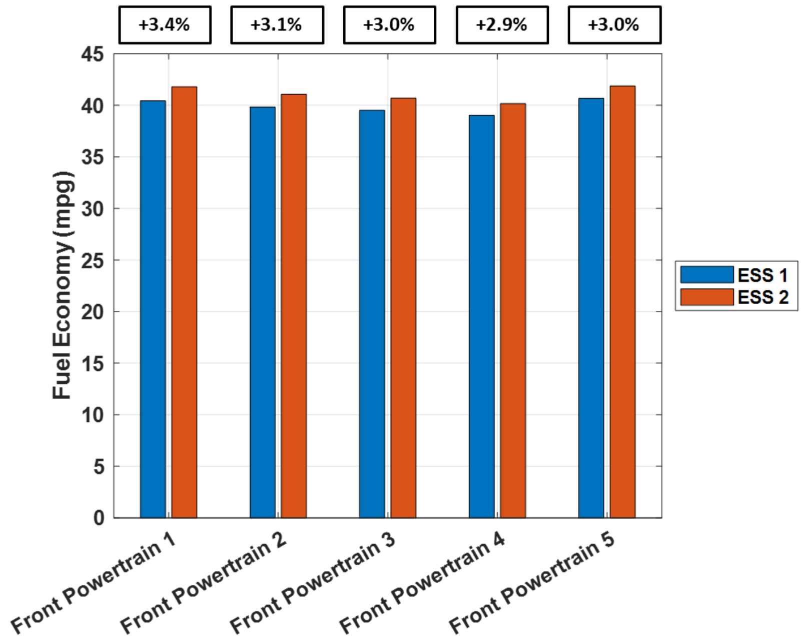


Figure 15: EMC Highway Fuel Economy Results for Varying ESS

The trends from the EMC highway drive cycle showed that ESS Option 2 had a fuel economy improvement around 3% for all front power train options. The overall combined fuel economy improvement was 5.4-7.8% based on the competition weighting of city and highway driving as shown in Figure 16.

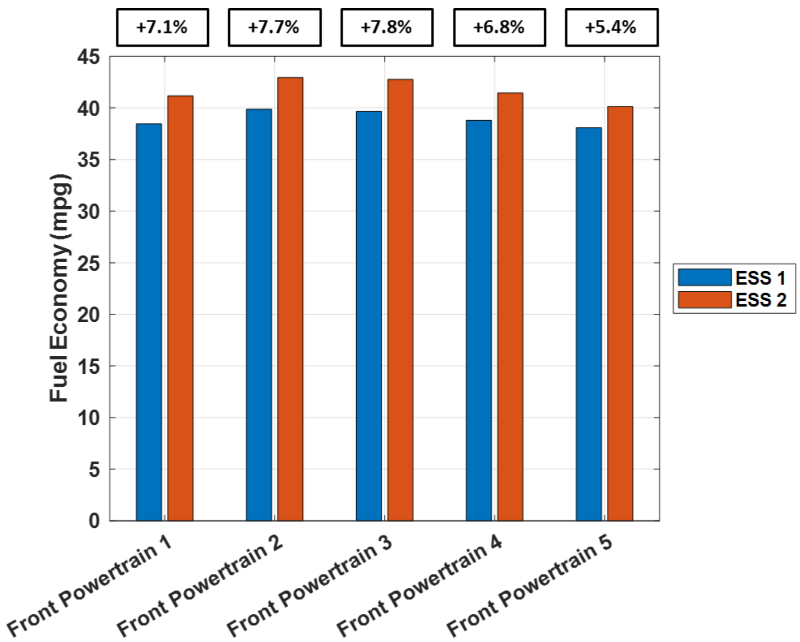


Figure 16: EMC Combined Fuel Economy Results for Varying ESS

Both ESS had similar mass and packaging concerns. However, ESS 2 saw a benefit in the combined fuel economy measurement and was chosen as the best available ESS solution for this hybrid system with the goal of minimizing fuel consumption.

### 4.3.2. Front Powertrain

There were five front powertrain systems evaluated found in Table 5.

Table 5: Front Powertrain System Evaluated

| Front Powertrain Options | GM Engines | Front Final Drive Ratio |
|---|---|---|
| Front Powertrain Option 1 | LVG 1.5L Turbo | 3.49 |
| Front Powertrain Option 2 | LTG 2.0L Turbo | 2.89 |
| Front Powertrain Option 3 | LTG 2.0L Turbo | 3.17 |
| Front Powertrain Option 4 | LTG 2.0L Turbo | 3.8 |
| Front Powertrain Option 5 | LCV 2.5L NA | 3.47 |

The final dive ratio of 8, the rear electric motor of 112 kW, and ESS with 3.5 kW-hr capacity were held constant throughout this analysis. The front powertrain options were varied to consider various engines and front final drive ratios. There were various 8 speed transmission linked to the different engines, however each transmission had the same gear ratios. The fuel economy trends for the different front powertrains can be found in

Figure 17.

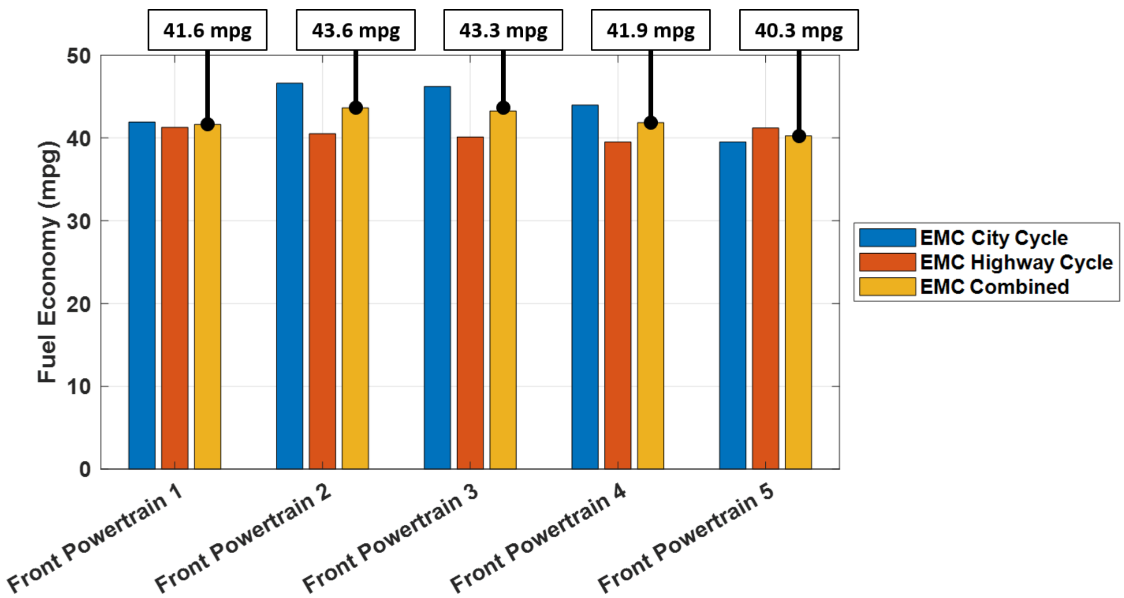


Figure 17: Fuel Economy Results for Varying Front Powertrains

The front powertrain systems combined fuel economy are within 3.3 mpg showing a small fuel economy benefit for the front powertrain systems with the 2.0L turbo engine.  The 2.0L turbo showed a major benefit for city driving due to the ability to manipulate the torque split to favor slightly charging up the ESS and then having longer engine off periods. The final front powertrain system was front powertrain 3. Front powertrain 3 had benefits outside of these simulations that gave it the edge over front powertrain 2. Front powertrain 3 was the 2.0L turbo and 3.17 front final drive ratio as well as a transmission that had an accumulator and electric transmission range shifter (ETRS). The accumulator will help during start stop operations and the electric transmission range shifter will allow for more control over the transmission.

### 4.3.3. Rear Electric Motor

When choosing the rear electric motor, two options were evaluated with the specifications found in Table 6.

Table 6: Rear Electric Motor Evaluated

| Component Properties | Motor Option 1 | Motor Option 2 |
|---|---|---|
| Part Number | GVM210-100 | GVM210-150 |
| Supplier | Parker Hannifin | Parker Hannifin |
| Peak Power/ Peak Torque | 90 kW/ 168 N-m | 112 kW/ 258 N-m |
| Continuous Power/ Continuous Torque | 57 kW/ 95 N-m | 83 kW/ 148 N-m |
| Winding | 2 Stack | 3 Stack |

Various other motor sizes were considered in an initial design space search with the vehicle model that utilized dynamic programming. Smaller motors with a peak power between 30-80 kW were unable to fully capture the available regenerative braking torque as well as the smaller motors could not provide ample torque and required the engine to be turned on to meet the full drivers request. The two motors being evaluated had their torque capabilities shown in Figure 18 and Figure 19. This simulation only considered the continuous power and torque limits.  Motor 2 was able to meet the drivers torque request for the city and highway drive cycles without the engine. This led to Motor 2 having a higher number of viable control variable combination. Both motors had a powered motor speed limit of 8000 RPM and a free-spinning motor of 8500 RPM.

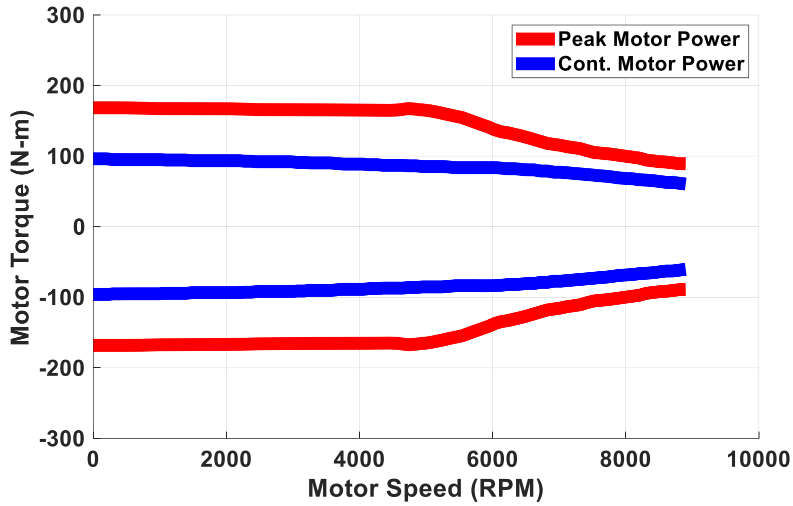


Figure 18: Motor 1 Torque Map

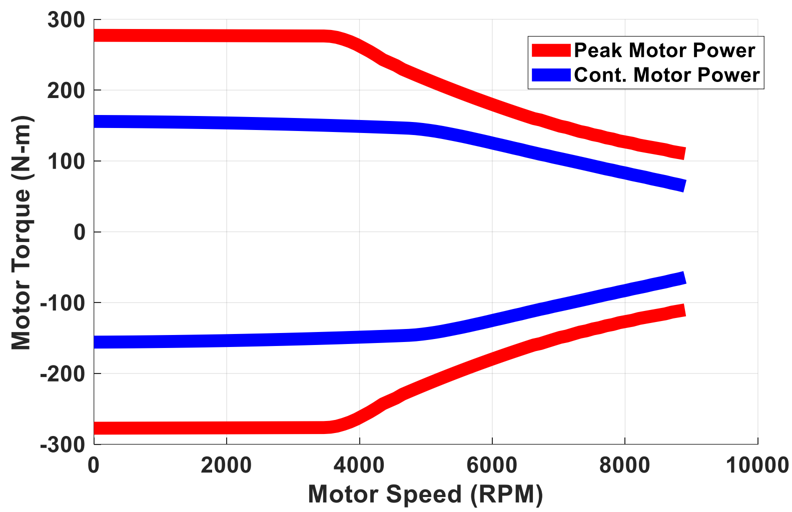


Figure 19: Motor 2 Torque Map

For the presented simulations, the rear drive ratio of 8, the option 3 front powertrain system, and the option 2 ESS were held constant. For this analysis, the only parameter that was varied was the rear electric motor with the results shown in Figure 20.

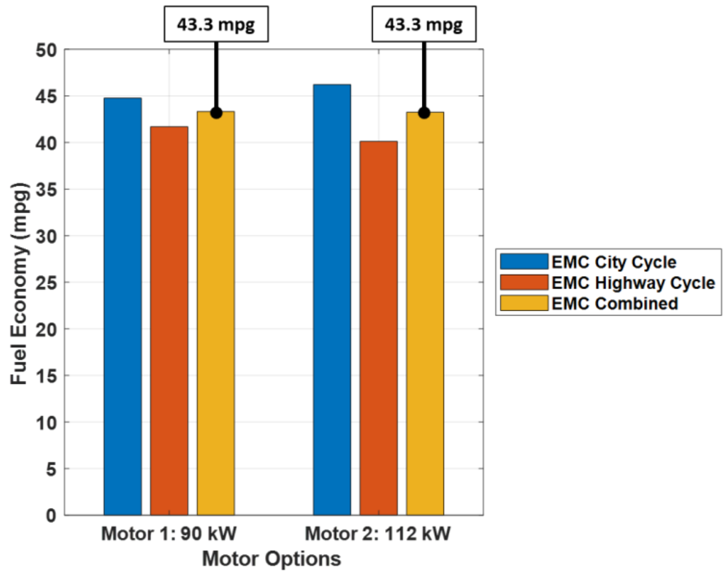


Figure 20: Fuel Economy Results for Varying Rear Electric Motor

The results showed that the combined fuel economy with each motor is estimated to be the same. However, Motor 1 performs better during the highway driving, and Motor 2 shows an advantage during city driving. Both motors had similar packaging challenges, and a negligible mass difference. Motor 2 was chosen due to the expected additional city look ahead information for vehicle to everything (V2X) optimization and the available full EV mode. However, this tradeoff was made understanding the lower motor efficiency during highway driving conditions.

### 4.3.4. Rear Final Drive Ratio

When choosing the rear final drive ratio, four options were evaluated with the specifications found in Table 7. These final drive options were available with the BorgWarner eGearDrive. Additional ratios of 8.76 and 9.00 were available, however they were eliminated based on vehicle top speed requirements and the motor speed limits.

Table 7: Rear Final Drive Ratios Evaluated

| Options | Rear Final Drive Ratios |
|---|---|
| Option 1 | 6.54 |
| Option 2 | 7.17 |
| Option 3 | 8.00 |
| Option 4 | 8.26 |

When evaluating these final drive ratios, the rest of the components were held constant based on the decisions discussed in Section 4.3.1-Section 4.3.3. The fuel economy results can be found in

Figure 21 for the varied rear final drive ratios.

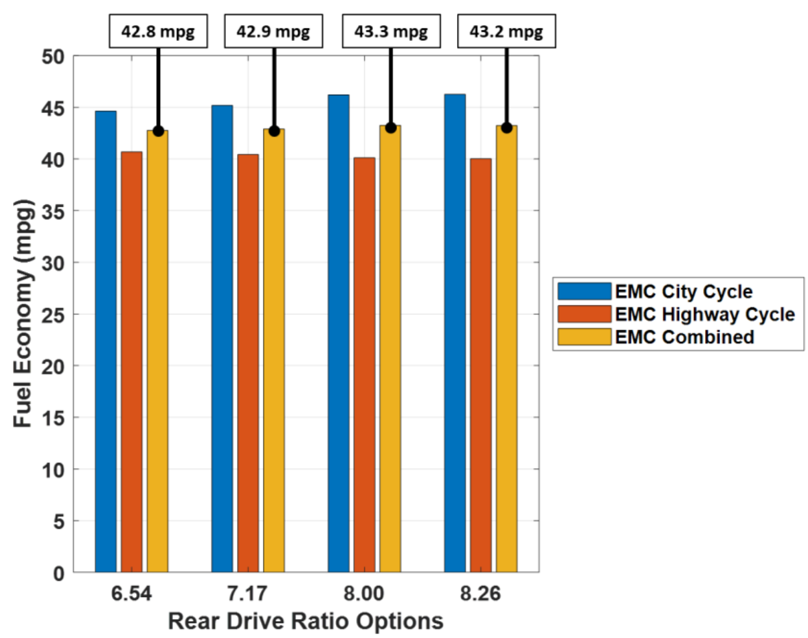


Figure 21: Fuel Economy Results for Varied Rear Drive Ratios

When varying the rear final drive ratio, there was a slight fuel economy benefit for higher rear drive ratios. The rear drive ratio of 8.00 was selected, due to the fuel economy benefit and the ability to exceed the competition vehicle top speed requirement by more than 10 mph.

4.3.5. Chapter Summary

This chapter began by discussing the process of determining the proper discretization required to run simulations that converged to the minimum fuel consumption for each component combination. Then, the chapter reviewed the simulations that were performed, and the specific trends associated with sizing each of the hybrid vehicle components. The final vehicle architecture the team chose can be found in Figure 22.

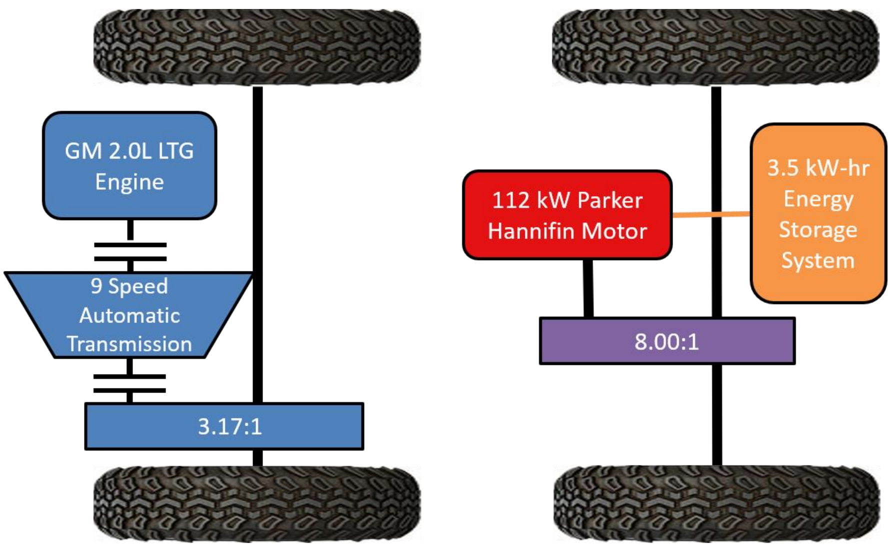


Figure 22: Final Vehicle Architecture Selected

# Chapter 5. Conclusion

This research successfully met its three major objectives:

- To ensure that the model accurately represent the vehicle behavior with proper implementation of component data.
- To ensure that control algorithm acts in a feasible manner, and meets the competition specified drive cycles.
- To determine the optimal size of a rear electric motor (REM), rear final drive ratio, engine, transmission, and energy storage system (ESS) to minimize fuel consumption.

The finalized hybrid vehicle architecture can be found in Figure 22. These components will create a through the road hybrid electric vehicle. There are four major advantages for this vehicle architecture:

1. An all-electric mode capable of meeting all torque request associated with the EMC city and EMC highway drive cycle.
2. The ability to capture all available regenerative braking energy.
3. A higher-powered engine with regions of higher efficiency compared to other engines.
4. A higher rear final drive ratio that allows the motor to operate more efficiently.

These benefits will be further explored as the vehicle develops over the entirety of the EcoCAR Mobility Challenge. Future work associated with utilizing dynamic programming include rules extraction and real time implementation. After running various drive cycles, rules will be extracted to govern the operation mode, gear state, and torque split. This will lay the foundation for a rules-based controller. Later in this competition, teams are expected to optimize the fuel economy with vehicle to everything (V2X) information. By knowing an optimization horizon, dynamic programming can be real time implementable. Within research, this technology has obtained 15-20% fuel economy improvement [8]. In conclusion, the optimization strategy known as dynamic programming was utilized to design a hybrid vehicle within the competition requirements that maximizes fuel economy.

# Chapter 6. Bibliography


[1] J. R. Josephson, B. Chandrasekaran, M. Carroll, N. Iyer, B. Wasacz, G. Rizzoni, Q. Li and D. Erb, "An Architecture for Exploring Large Design Spaces," in *Proc. Nat. Conf. American Association for Artificial Intelligence*, Madison, WI, 1998.

[2] D. T., S. L. and R. G., "A two-step optimization method for the preliminary design of a hybrid electric vehicle," *International Journal of Electric and Hybrid Vehicle,* vol. 1, no. 2, pp. 142-165, 2008.

[3] S. Onori, L. Serrao and G. Rizzoni, Hybrid Electric Vehicles Energy Management Strategies, London: Springer-Verlag London Ltd, 2016.

[4] R. E. Bellman, Dynamic Programming, Princeton: Priceton Unversity Press, 1957.

[5] D. A. Q. Z. X. a. R. G. Jung, "Mission-based Design Space Exploration for Powertrain Electrification of Series Plugin Hybrid Electric Delivery Truck," in *SAE Technical Paper*, Detriot, 2018.

[6] O. Sundstrom and L. Guzzella, "A generic dynamic programming Matlab function," *IEEE Control Applications, (CCA) & Intelligent Control, (ISIC),* pp. 1625-1630, 2009.

[7] Ohio Supercomputer Center, *Ohio Supercomputer Center,* Columbus OH: Ohio Supercomputer, 1987.

[8] P. A. K. T. L. C. K. e. a. Olin, "Reducing Fuel Consumption by Using Information from Connected and Automated Vehicle Modules to Optimize Propulsion System Control," in *SAE Technical Paper*, 2019.